REMARKS

The Office Action mailed on May 13, 2009 has been reviewed. Claim 9 has been amended. Claims 1-5, 8-13, 16-21, and 23 are pending in this application.

Rejections Under 35 U.S.C. § 101

Claims 9-13 and 16 were rejected under 35 USC § 101 because the claimed invention is directed to non-statutory subject matter.

In order to expedite prosecution, claim 9 has been amended to address the alleged issue noted in the Office Action.

Accordingly, Applicant respectfully requests that this rejection be withdrawn.

Rejections Under 35 U.S.C. § 102

Claims 1-2, 5, 8-10, 13, 16-18, and 23 were rejected under 35 USC § 102(e) as being anticipated by Dailey, (U.S. Publication No. 2003/0217093).

Claim 1 of the present application recites, in part, "determining when one of the plurality of periodic events occurs; determining, for each of the set of services associated with that periodic event, if that service is enabled for execution; and distributing the execution of the services associated with that periodic event that are enabled throughout a next periodic interval of time associated with that periodic event following the occurrence of that periodic event."

The Office Action took the position that "determining, for each of the set of services associated with that periodic event, if that service is enabled for execution" is taught in paragraphs [0028]-[0031] of Daily. The Office Action asserted that this language teaches "when the periodic event occurs, based on the bit associated with the task, the task manager determining which task is enable/ready for execution".

Applicant traverses the Office Action's characterization of the operation of Dailey. The Office Action appears to be referring to the operation of the service variable

59 of Dailey. The service variable 59 of Dailey is used to determine when a particular periodic event has occurred using a single timer variable. This is clearly shown in the flow charts (FIGS. 2 and 3) of Dailey. As can be seen in FIG. 2, bits of the service variable 59 are set based on *the state of the timer variable 58*, not based on whether a particular periodic task has been enabled for execution. The processing of FIG. 2 of Dailey is part of an interrupt handler that is executed on each periodic tick interrupt. As shown in FIG. 3 of Dailey, in block 40 of Dailey, the task manager initially clears the service variable 59 so that the bits of the service variable 59 are set only by the interrupt handler. Then, the task manager checks if the "tick expired" variable has been set by the interrupt handler in connection with performing the processing of FIG. 2 (block 42). Then, the task manager cycles through the bits of the service variable 59 to determine if it is time to execute the various periodic tasks. This determination is made based on whether the bit of the service variable 59 associated with each periodic task is set. If the associated bit is set, it is time to execute the associated particular periodic task. It is noted that FIG. 3 clearly shows that the associated bit of the service variable 59 is cleared each time the associated periodic task is executed. (See, e.g., blocks 47, 50, and 53). Thus, *the service variable 59 is used as a single timer variable for all of the periodic tasks and is not used for enabling or disabling execution of the periodic tasks.* See, e.g., Dailey, paragraph [0008] ("Technical advantages of certain embodiments of the present invention include the ability to use a single service variable to maintain a timer for multiple processing tasks, instead of using several computer variables.").

Dailey is silent as to any mechanism for enabling and disabling the execution of any of the periodic processing tasks (and the checking for such enabling or disabling). Indeed, Dailey indicates that the schedule of such tasks does not change. See, e.g., Dailey, paragraph [0020] ("Typically, microcontroller 20 utilizes task manager 22 to manage specific tasks retained in memory. Generally, these tasks are performed at various times and the schedule for performing the tasks usually does not change. Specifically, task manager 22 may manage these tasks to be performed at given intervals.").

In response to substantially the same arguments set forth in Applicant's most recent Appeal Brief, the Office Action stated the following:

> Examiner respectful disagreed with applicant. First, examiner noted that the recited limitation of "determining, for each of the set of services associated with that periodic event if that service is enabled for execution" did not suggested the function of enabling and disabling the execution of the periodic tasks. Second, Dailey teaches in (paragraphs [0028]-[0031]) when the periodic event occurs, based on the bit associated with the task, the task manager determining which task is enable/ready for execution is the same as determining if that service is enabled (ready) for execution for each of the set of services associated with that periodic event. Therefore, Dailey clearly teach the recited limitation.

With respect to the statement that "*First, examiner noted that the recited limitation of "determining, for each of the set of services associated with that periodic event if that service is enabled for execution" did not suggested the function of enabling and disabling the execution of the periodic tasks.*", Applicant notes that Applicant's argument is that Dailey fails to teach the claim language "determining, for each of the set of services associated with that periodic event, if that service is enabled for execution". One reason (of many) for this conclusion is that Dailey does not include any mechanism for enabling or disabling execution of the periodic tasks. That is why Dailey does not teach or suggest "determining, for each of the set of services associated with that periodic event if that service is enabled for execution" — there is no way to disable or enable a periodic task so Dailey does not check (and has no reason to check) if a periodic task is enabled.

With respect to the statement that "*Second, Dailey teaches in (paragraphs [0028]-[0031]) when the periodic event occurs, based on the bit associated with the task, the task manager determining which task is enable/ready for execution is the same as determining if that service is enabled (ready) for execution for each of the set of services associated with that periodic event.*", the Office Action appears to be arguing that

"determining ... if [a] service is *enabled* for execution" is the same as determining if a service is *ready* for execution. The Office Action does not provide any evidence or explanation as to why that is this case. Applicant respectfully submits that "determining ... if [a] service is *enabled* for execution" is *not* the same as determining if a service is *ready* for execution. For example, in embodiments of the subject matter recited in claim 1 of the present application, a particular service could be *ready* to execute but not *enabled* for execution.

Also, contrary to what appears to be asserted in the Office Action, paragraphs [0028]-[0031] of Dailey do not actually teach determining if a periodic task is *ready* for execution. Instead, each bit of the service variable 59 described in paragraphs [0028]-[0031] of Dailey is used for indicating ***when*** to execute a task—it provides no indication whatsoever that the particular task is somehow "ready" to be executed. As noted above, the service variable 59 of Dailey is used to maintain a timer for multiple processing tasks using a single variable instead of using multiple variables. *See, e.g.,* Dailey, paragraph [0008]. In other words, the service variable 59 of Dailey is just a timer.

Claims 2, 5 and 8 depend from claim 1. Therefore, at least the arguments set forth above with respect to claim 1 apply to these dependent claims. Applicant, however, does not concede any assertion made in the Office Action with respect to these dependent claims and reserves the right to provide additional arguments directed to the dependent claims if a further response is required.

It is respectfully submitted that at least the arguments set forth above with respect to claim 1 apply to the rejections of claims 9-10, 13, 16-18, and 23. Applicant, however, does not concede any assertion made in the Office Action with respect to these claims and reserves the right to provide additional arguments directed to the claims if a further response is required.

Accordingly, Applicant respectfully requests that this rejection be withdrawn.

Rejections Under 35 U.S.C. § 103

Claims 3, 11, and 19 were rejected under 35 USC § 103(a) as being unpatentable over Dailey, (U.S. Publication No. 2003/0217093).

Claim 3 depends from claim 1. Applicant respectfully submits that at least the arguments set forth above with respect to claim 1 apply to claim 3 as well.

Moreover, claim 3 further recites "further comprising configuring at least one of the set of services associated with that periodic event in a one-shot mode in which the service is enabled for execution one time and then is disabled."

The Office Action conceded that Dailey does not specifically teach this language from claim 3. The Office action took the position that:

> However, Dailey disclosed setting a execution trigger in a service variable at a predetermined time interval, and task manager based on the service variable enable only one task of a set of processing tasks to perform at each interval (paragraphs [0005], [0035], page 5, claim 14).

The Office Action then concluded that:

> It would have been obvious to one of an ordinary skill in the art at the time the invention was made to have recognized that Dailey indirectly configuring (setting) at least one of set services (one task of a set of processing task) to perform only one time in the predetermined interval is obvious as configuring at least one set of services associated with that periodic event in a one-shot mode in which the service is enable for execution one time of the claimed invention. In addition, since the task only allow to perform one per interval, thus after the interval is reached, it would have been obvious for the task to disable. Therefore, it would have been motivated to one of an ordinary skill in the art at the time the invention was made to use the teaching of Dailey to distribute periodic task workloads in such a way that allow the system to process other processing work such as real-time tasks (Dailey, paragraph [0009]

Applicant respectfully traverses this rejection. Dailey explicitly relates to **periodic** tasks (that is, tasks that occur once every period). The "execution trigger" referred to in Dailey is a bit within the service variable described above and is simply used to implement multiple-timers using a single variable instead of multiple variables. Moreover, as noted above, Dailey does not teach enable or disabling of periodic tasks as alleged in the Office Action.

The Office Action rejected claims 11 and 19 for the same reasons as claim 3. Therefore, Applicant respectfully submits that at least the arguments set forth above with respect to claim 3 apply to claims 11 and 19 as well.

Accordingly, Applicant respectfully requests that this rejection be withdrawn.

Claims 4, 12, and 20-21 were rejected under 35 USC § 103(a) as being unpatentable over Dailey, (U.S. Publication No. 2003/0217093) and in view of Nakano et al. (U.S. Patent No. 7,039,012).

Applicant respectfully submits that at least the arguments set forth above with respect to claim 1 apply to these rejected dependent claims. Applicant, however, does not concede any assertion made in the Office Action with respect to these dependent claims and reserves the right to provide additional arguments directed to the dependent claims if a further response is required.

Accordingly, Applicant respectfully requests that this rejection be withdrawn.

CONCLUSION

Applicant respectfully submits that claims **1-5, 8-13, 16-21, and 23** are in condition for allowance and notification to that effect is earnestly requested. If necessary, please charge any additional fees or credit overpayments to Deposit Account No. 502432.

If the Examiner has any questions or concerns regarding this application, please contact the undersigned at the telephone number listed below.

Respectfully submitted,

Date: 2009-08-12

/Jon M. Powers/
Jon M. Powers
Reg. No. 43868

Attorneys for Applicant
Fogg & Powers LLC
5810 W. 78th Street, Ste. 100
Minneapolis, MN 55439
T – (952) 465-0770
F – (952) 465-0771